IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 2, 2019, accepted September 23, 2019, date of publication October 7, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945889

A Distributed Approach for High-Dimensionality Heterogeneous Data Reduction

RANIA MKHININI GAHAR[1], OLFA ARFAOUI[1], MINYAR SASSI HIDRI[2], AND NEJIB BEN HADJ-ALOUANE[1]

[1]OASIS Research Lab, National Engineering School of Tunis, University of Tunis El Manar, Tunis 1068, Tunisia
[2]Computer Department, Deanship of Preparatory Year and Supporting Studies, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

Corresponding author: Rania Mkhinini Gahar (rania.mkhininigahar@enit.rnu.tn)

ABSTRACT The recent explosion of data size in number of records and attributes has triggered the development of a number of Big Data analytics as well as parallel data processing methods and algorithms. At the same time though, it has pushed for usage of data Dimensionality Reduction (DR) procedures. Indeed, more is not always better. Large amounts of data might sometimes produce worse performance in data analytics applications, and this may be caused by the presence of missing data. These latter are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. In this work, we propose a new distributed statistical approach for high-dimensionality reduction of heterogeneous data that is based on the MapReduce paradigm, limits the curse of dimensionality and deals with missing values. To handle these latter, we propose to use the Random Forest imputation's method. The main purpose here is to extract useful information and reduce the search space to facilitate the data exploration process. Several illustrative numeric examples using data coming from publicly available machine learning repositories are also included. The experimental component of the study shows the efficiency of the proposed analytical approach.

INDEX TERMS Heterogeneous data, missing data, curse of dimensionality, big data, mapreduce, descriptive analysis, feature selection.

I. INTRODUCTION

Big data is the aggregation of large-scale, voluminous, and multi-format data streams originated from heterogeneous and autonomous data sources [40]. The volume is the primary characteristic of big data that is represented by the acquisition of storage spaces in large-scale data centers and storage area networks. The massive size of the big data not only causes the data heterogeneity but also results in diverse dimensionalities in the datasets. Therefore, efforts are required to reduce the volume to effectively analyze big data [6].

For about thirty years, data analysis methods have largely demonstrated their effectiveness in data processing in many fields. Data reduction is one of these methods and part of the descriptive (or exploratory) analysis. It tries to summarize a sample of data using graphs or numerical characteristics. This implies that data reduction is part of the multivariate exploratory statistics which seek to reduce the number of data dimension by extracting a number of factors, components, clusters, etc., which explain the dispersion of (multidimensional) data.

In the case where it is certain that data (which is in most cases voluminous) contains useful information, the reduction methods can be applied in order to have a synthetic view and so to make easy the exploration process. The issue comes down to a problem of data structuring and knowledge extraction.

In this context, the simultaneous introduction of both quantitative and qualitative variables in a dataset for analysis becomes a frequent problematic when we talk about reducing heterogenous data. In fact, and when we talk about quantitative variable, we refer to mathematical variables measured in numerical quantities. For example, in the physical field, distance, mass, time and wave frequency are quantitative variables, in the social sciences, the age of a population is a quantitative variable, in psychology, the intelligent quotient is also a quantitative variable. When we refer to qualitative variables (called also categorical variables), we involve the measurement of the element's quality. The color of hair (e.g, black, white, red) or the type of the music (classic, slow,

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Kashif Bashir.

rock) are examples of nominal categorical variables. We distinguish also the special case of ordinal variables which are numerical qualitative variables. For example, a customer appreciation for a product can be noted from -2 (very bad) to $+2$ (excellent), going through zero (indifferent).

In the context of multidimensional data reduction, it is very frequent for the variables to be of a mixed nature (quantitative and qualitative). The usual processing consists in either putting the qualitative variables (resp. quantitative) into illustrative elements in a Principal Component Analysis (PCA) [37] (resp. Multiple Correspondence Analysis (MCA) [38]), or discretizing the quantitative variables into qualitative variables according to an MCA. Which very often introduces a bias due to the choice of the number of classes and their amplitudes equal or different, and which causes a loss of information.

Many researchers are interested in this issue and have proposed methods that deal simultaneously with the two types of variables into active elements: the PCA with indicators [37] and more recently the Factor Analysis of Mixed Data (FAMD) [29]. But when it comes to high-dimensional data systems which aggregate massive data streams from heterogeneous data sources, many questions are asked and the most important ones are:

- Is data reduction a serious challenge in high-dimensional data?
- What the are the techniques or solutions to cater the effect of reducing high-dimensional data without effecting the data value?

In fact, the methods of reducing the data representation space size can be processed either by the extraction or selection of the attributes. The extraction of the attributes transforms the initial attributes' space into a new one formed by the linear or non-linear combination of the initial attributes. Contrariwise, the selection of attributes selects the most relevant attributes according to a given criterion.

In these directions, and to answer the previous questions, we propose in this work to prove that the high-dimensionality reduction is really a serious challenge, and this, by suggesting a new distributed statistical approach that reduces the huge amount of heterogeneous data without affecting their value.

The rest of this paper is organized as follows: the Big Data complexity and the need to limit the curse of dimensionality is presented in section 2. Section 3 presents some statistical methodologies for descriptive data analysis. Section 4 introduces the proposed approach for reducing high-dimensional heterogenous data with missing values imputation. Section 5 discusses the computational results. Overall discussion with future extension remarks are stated in section 6.

II. ON BIG DATA COMPLEXITY AND THE DIMENSIONALITY CURSE LIMIT NEED

Research on Big Data analytics is entering in the new phase called fast data where multiple gigabytes of data arrive in the Big Data systems every second. Modern Big Data systems collect inherently complex data streams due to the 3 basic Vs [21] which are Volume, Velocity and Variety and to which added Veracity, Validity, Vulnerability, Volatility, Visualization and Value [27] of Big Data.

The well-designed big data systems must be able to deal with these dimensions effectively by creating a balance between data processing objectives and the cost of data processing (i.e., computational, financial, programming efforts) in big data systems. Data collection and storage capabilities have enabled researchers in diverse domains to observe and collect huge amount of data. However, large datasets present substantial challenges to existing data analysis tools. One major bottleneck in this regard is the large number of features or dimensions associated with some measured quantity. This problem frequently termed as ''the curse of dimensionality'' and deals with the cost of performing a reliable analysis that increases exponentially with dimension (variable, feature, attribute).

In addition, existing algorithms usually scale very poorly with the increase in number of the data dimensions. One solution to this problem is mapping the data from the high dimension space to a lower one while preserving almost the original structure of the data.

Dimensionality Reduction (DR) in general is the process of converting a set of data having high dimensions into data with lesser/lower dimensions ensuring that it conveys similar information concisely without affecting original information. DR can be considered a problem of global combinatorial optimization in machine learning, which reduces the number of features, removes irrelevant, noisy and redundant data, to obtain the accuracy and saves the computation time and simplifies the result. In fact, when it comes to Big Data, the analysis'process has been a great difficulty with number of dimensions (attributes) that adds much more difficulty to study the data.

To describe the DR, we assume that we have dataset represented in a $n \times p$ matrix y of n data vectors y_i with $i \in \{1, 2, \ldots, n\}$ with dimensionality p. We assume further that this dataset has intrinsic dimensionality k (where $k < p$, and often $k \ll p$). Mathematically, intrinsic dimensionality means that the points in dataset y are lying on or near a manifold with dimensionality k that is embedded in the p dimensional space. Dimension reduction techniques transform dataset y with dimensionality p into a new dataset x with dimensionality k, while retaining data geometry as much as possible [1], [5], [10], [11], [19].

The whole DR process is divided into two parts:

- ***Feature Selection*** also known as Variable Selection, Attribute Selection and Variable Subset Selection. It is a process of selecting the most important features (variables, attributes) for building models. By using feature selection technique we can remove the redundant or irrelevant features from the data without much loss of information.
- ***Feature Extraction***: the transformation from high-dimensional data to reduced feature data is called as

Feature Extraction. The extracted feature are expected to be relevant information from the given input dataset.

A wide range of DR methods are proposed in the existing literature. Some of the recent techniques with their features and other performance measures are presented in the table 4.

In recent years, life sciences have undergone a tremendous revolution following the rapid development of advanced technologies and laboratory instruments. The biomedical field may be the best example that has seen a remarkable advance since the advent of complete sequences of the genome. This post-genomics era has led to the development of new high-throughput techniques that are generating enormous amounts of data. These later may have much more variables than observations. In fact, research shows that fuzzy classification methods are a good choice for dealing with the above-mentioned problems. Recently, the authors of [36] presented linguistic hedges fuzzy classifier with selected features (LHNFCSF) to reduce dimensions, select features and perform classification operations by discarding redundant, noise-corrupted or even unimportant features. The results indicated that LHNFCSF not only help reducing the dimensionality of large medical data sets but also can speed up the learning process and improve the classification performance. Not far to this field, Bioinformatics research involves large volumes of data and complex data analytics. Most of the tools in bioinformatics use iterative machine learning methods. These tools can be scaled to handle large volume of data by using parallel and distributed computing models as provided in Hadoop and Spark. Big data tools and platform use more different data models and formats than used by traditional bioinformatics tools. The authors in [4] explore the data formats used by different tools and algorithms and also presents modern data formats that are used on Big Data Platform. It will help researchers and developers in choosing appropriate data format to be used for a particular tool or algorithm.

Single-cell RNA-seq data allows insight into normal cellular function and various disease states through molecular characterization of gene expression on the single cell level. Dimensionality reduction of such high-dimensional data sets is essential for visualization and analysis, but single-cell RNA-seq data are challenging for classical dimensionality-reduction methods because of the prevalence of dropout events, which lead to zero-inflated data. That's why, the authors of [30] develop a dimensionality-reduction method, (Z)ero (I)nflated (F)actor (A)nalysis (ZIFA), which explicitly models the dropout characteristics, and show that it improves modeling accuracy on simulated and biological data sets. One of the limitations of ZIFA is that it models strictly zero measurements rather than near-zero values. It has been possible to account for near-zero values in a univariate mixture modeling framework by placing a small-variance distribution around zero rather than a point mass. Achieving the same goal, in a multivariate context, requires further methodological thought and development to produce solutions that are computationally tractable with a large number of dimensions.

PCA [39] is a powerful tool in DR for highly correlated data. But, when it comes to big data, classical PCA approaches cannot be applied because of memory and storage barriers. To solve the problem, the authors of [42] propose a new approach. The basic idea is to derive an array of sufficient statistics by scanning data by rows. It shows that the proposed approach can provide exact solutions if the linear regression approach is used in the follow up analysis. This is an important property in the general PCA approach for big data, which will have great impacts on future research on dimension reduction for big data.

Human activity recognition (HAR) is an emerging research topic in pattern recognition, especially in computer vision. The main objective of human activity recognition is to automatically detect and analyze human activities from the information acquired from different sensors. Human activity prediction using big data remains a challengingly open problem. Several approaches have recently been developed in order to find practical ways to solve high dimensionality of data problems. The authors of [11] propose a framework that deals with HAR modeling involving a significant number of variables in order to identify relevant parameters from data and thus to maximize the classification accuracy while minimizing the number of features.

Recently, the importance of mobile cloud computing has increased. Mobile devices can collect personal data from various sensors within a shorter period of time and sensor-based data consists of valuable information from users. Advanced computation power and data analysis technology based on cloud computing provide an opportunity to classify massive sensor data into given labels. Random forest algorithm is known as black box model which is hardly able to interpret the hidden process inside. The authors of [8] propose a method to measure the influence of variables using Shapley Value method in random forest algorithm [34]. The approach tries to solve the multicollinearity problem in other techniques.

The DR contributions evolve with time. The authors of [41] propose a new method named cumulative slicing principle fitted component (CUPFC) model to conduct sufficient DR and prediction in regression. Based on the classical PFC methods, the CUPFC avoids selecting some parameters such as the specific basis function form or the number of slices in slicing estimation. The simulations investigate the effectiveness of this method in prediction and reduction estimation with other competitors. The results indicate that the new proposed method generally outperforms the existing PFC methods no matter how the predictors are truly related to the response. The application to real data also verifies the validity of the proposed method.

The current data tends to be more complex than conventional data and need DR. This latter is important in cluster analysis and creates a smaller data in volume and has the same analytical results as the original representation.

A clustering process needs DR to obtain an efficient processing time while clustering and mitigate curse of dimensionality. In [9], authors propose a model for extracting multidimensional data clustering of health database. The results show that DR significantly reduce dimension and shorten processing time and also increased performance of cluster in several health datasets.

The wide spectrum view of the proposed methods for big data reduction uncovers that the DR is being carried out at several levels of big data architecture and in different forms.

III. STATICAL METHODOLOGIES FOR DESCRIPTIVE ANALYSIS

In this section, we present the most used statical methodologies for the descriptive data analysis. We focus on data preparation since it presents an important step in the data pre-processing which will be analyzed later.

Data cleaning or data preparation is an essential part of statistical analysis. In practise, it is often more time-consuming than the statistical analysis itself. 80% of data analysis is spent to clean and prepare data. This process consists in transforming raw data into consistent data that can be analyzed. It aims at improving the content of statistical statements based on the data as well as their reliability.

Data cleaning may profoundly influence the statistical statements based on the data. Typical actions like imputation or outlier handling obviously influence the results of a statistical analysis. For this reason, data cleaning should be considered as a statistical operation to be performed in a reproducible manner. Next, we detail the different steps of data cleaning.

A. EXPLORING RAW DATA

The first step in the data cleaning process is exploring raw data. We can think of data exploration itself as a three step process consisting in understanding the structure of our data, looking and finally visualizing it.

To understand the structure of the dataset, we check firstly the class of the data object to verify that it's a data frame, or a two-dimensional table composed of rows and columns, in which each column is a single data type such as numeric, character, etc.

B. TIDYING DATA

Tidying data is a small but an important aspect of data cleaning process which consists in structuring dataset to facilitate analysis. In fact, the principle of tidy data provides a standard way to organize data values which are qualified as messy, within a dataset. This step has been designed to facilitate initial exploration and data analysis and also to simplify the development of data analysis tools that work well together. Hence, in tidy data we have:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

Most statistical datasets are represented in the form of rectangular tables constructed from rows and columns. These tables are labelled in the most of time.

Let's distinguish the following data structure through pregnancy data sample. Table 1 provides a data sample with two columns and three rows. Each of them is labeled.

TABLE 1. Pregnancy data sample.

	#Treatment a	*#Treatment b*
John Smith	-	2
Jane Doe	16	11
Mary Johnson	3	1

Table 2 presents the same data sample as Table 1 but the rows and columns have been transposed. The data is the same, but the layouts are different.

TABLE 2. The same dataset structured differently.

	John Smith	*Jane Doe*	*Mary Johnson*
Treatment a	-	16	3
Treatment b	2	11	1

Tables 1 and 2 contain messy data because the vocabulary of rows and columns is not rich enough to say that it is the same data. So what is lacking in these two presentations is the semantics of the presented values.

A dataset is a collection of values, usually either quantitative or qualitative. Values are organized in two ways. Every value belongs to a variable and an observation. A variable contains all values that measure the same underlying attribute across units. An observation contains all values measured on the same units across attributes.

Table 3 presents the tidy version of the pregnancy data sample.

TABLE 3. The dataset with variables in columns and observations in rows.

Name	*Treatment*	*Result*
John Smith	a	-
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

We note here that the dataset contains 18 values representing three variables and six observations. The variables are:

1) *Name* with three possible values (John Smith, Mary Johnson, Jane Doe).
2) *Treatment* with two possible values (*a* and *b*).
3) *Result* with five or six values depending on how we think of the missing value (1,2,3,11,16,-).

Tidy data facilitate the extraction of needed variables to be used by an analyst or a computer because it provides a standard way to structure a dataset.

C. IMPUTATION OF MISSING DATA

It is very common to encounter missing data in statistics. However, the most of statistical methods can not be directly

TABLE 4. Dimension reduction methods.

Methods	*Description*	*Strengths*	*Weaknesses*
LHNFCSF [36]	Linguistic hedges neuro-fuzzy classifier with selected features.	Reduces the dimensions of the problem. Improves classification performance.	Lack of efficiency.
ZIFA [30]	DR for zero-inflated single-cell gene expression analysis.	Modeling the dropout characteristics. Improving modeling accuracy on simulated and biological datasets	Requires a combination of tools Increases the computational complexity. Models strictly zero measurements rather than near-zero values.
PCA for Big Data [42]	Deriving an array of sufficient statistics by scanning data by rows.	Supports parallelization High significance when used for quantitative Big Data.	Needs to handle qualitative Big Data Compromise on data quality.
Linear method for HAR [11] (Human Activity Recognition)	Using data mining techniques to deal with HAR modeling involving a significant number. of variables in order to identify relevant parameters from data and thus to maximize the classification accuracy while minimizing the number of features.	Better performance for DR (in terms of least time consumption and CPU expenditure).	Deals only with quantitative features.
Shapley Value method in random forest algorithm [8]	Clarify which variable affects classification accuracy to study variable impact. Prediction based on priority of the variables was obtained to study the classification result.	Tries to solve the multicollinearity problem in other techniques. Improves the accuracy of random forest prediction based on this priority.	Significant complexity.
DR for Bioinformatics data [4]	Explores the data formats used different tools and algorithms and also presents modern data formats that are used on Big Data platform.	Selects a data format that takes less space. and provides great performance.	Relates only to Bioinformatics data. Data Format complexity is substantial.
CUPFC [41]	Based on the classical PFC methods, the CUPFC avoids selecting some parameters such as the specific basis function form or the number of slices in slicing estimation	Conduct sufficient DR and prediction in regression.	Approach complexity is substantial.
DR before clustering [9]	Applying DR before clustering process.	Reduced dimension. Shorten processing time. Increased performance of cluster in several health datasets. Shows an increased cluster performance.	Experimental used datasets are not Big.

applied to an incomplete set of data because missing data can create problems for analyzing data. In this case, imputation is seen as a way to avoid problems involved with missing data.

Missing data can be replaced by plausible values, and this is called *simple imputation*. Nevertheless, simple imputation does not allow for uncertainty related to imputed data. To reflect this uncertainty, several imputations can be proposed for each missing data item. This is called *multiple imputation*.

In statistics, missing data (or missing values) occur when no data value is stored for the variable in an observation. Missing data are ambiguous and occur for a number of reasons such as individuals who do not answer to items of a questionnaire because of non or poor understanding, machines that fail, data which are considered unimportant at the time of entry, Inconsistencies with other data, and therefore data are deleted, etc.

1) MISSING DATA'S TYPES

In [24], the authors presented a typology which divide missing data into three categories:

- Missing Completely At Random (MCAR): a missing data is distributed in a completely random manner if the probability of absence is the same for all the observations. This probability depends, therefore, only on external parameters independent of this variable.
  For example, a participant flips a coin to decide whether to complete the depression survey. We note that if the amount of MCAR data is not important then ignoring cases with missing data will not bias the analysis. However, a loss of precision in the results is to be expected.
- Missing At Random (MAR): given the observed data, they are missing independently of unobserved one. For example, male participants are more likely to refuse to fill out the depression survey, but it does not depend on the level of their depression.
- Missing Not At Random (MNAR): missing observations are related to values of unobserved data. For example, participants with severe depression, or side-effects from the medication, were more likely to be missing at end.

2) MISSING DATA DISTRIBUTION

Let consider $Y = (y_{ij}) \in \mathbb{R}^{n\times p}$ the data rectangular matrix for p variables $Y_1, \ldots, Y_p$ and n observations. Consider $M = (m_{ij})$ the indication matrix of the missing values, which will define the distribution of the missing data. We will then consider 3 types of distributions:

- Univariate missing values: this means that for a variable Y_k only, if a y_{ki} observation is missing, then there will be no observation of this variable. An illustration is given in figure 1(a).

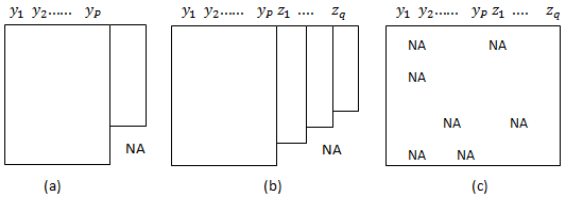

FIGURE 1. Distribution of missing values.

- Monotone missing values: the missing values are called monotone if Y_j is missing for an individual i. It implies that all the following variables $Y_k, k > j$ are missing for this individual (see figure 1(b)). The missing data indicator M is then an integer $M \in (1, 2, \ldots, p)$ for each individual, indicating the largest j for which Y_j is observed.
- Non-monotonic (or Arbitrary): in this case, the matrix of missing values is defined as $M = (m_{ij})$ with $m_{ij} = 1$ if y_{ij} is missing and zero otherwise (see figure 1(c)).

3) ABSENCE'S PROBABILITY

Suppose our data is $Y = (y_{ij}) \in \mathbb{R}^{n \times p}$, a rectangular matrix for p variables $Y_1, \ldots, Y_p$ and n observations. Consider $M = (m_{ij})$ the indication matrix of the missing values, the same size as Y which will define the distribution of the missing data.

The probability of absence according to the type of data missing (MCAR, MAR or MNAR) can then be expressed as a function of M [25]. The data are divided in two subsets according to the matrix M indicating the missing data. Such a random matrix is basically a collection of indicator variables with the property that y_{ij} is observed if $m_{ij} = 0$, and y_{ij} is missing whenever $m_{ij} = 1$.

Y and M together determine a partition of the data into two components : an observed and a missing component which we formally denote by $Y = \{Y_{obs}, Y_{mis}\}$.

We thus define $Y_{obs} = Y1_{\{m_{ij}=0\}}$ the observed data and $Y_{mis} = Y1_{\{m_{ij}=1\}}$ the missing data. The missing data mechanism is characterized by the conditional distribution of M knowing Y given by $p(M|Y)$.

In the case of MCAR data, the data absence (missingness) does not depend on the values of Y (Observed and unobserved values). We can cite for example :

- The case of flipping a coin to determine whether an observation goes missing.
- The probability of a birthweight being missing does not depend on
  - Observed data (child gender, smoking status, etc.)
  - Unobserved data (low birthweight does not influence risk of observation being missing)

$p(M|Y)$ is given as follows:

$$p(M|Y) = p(M) \, for \, all \, Y. \tag{1}$$

Consider now the case of MAR and let Y_{obs} be the observed part of the dataset and Y_{mis} the missing data. MAR means that the data absence depends only on Y_{obs} (Missingness unrelated to unobserved values).

For example, take political opinion polls. Many people refuse to answer. If you assume that the reasons people refuse to answer are entirely based on demographics, and if you have those demographics on each person, then the data is MAR. It is known that some of the reasons why people refuse to answer can be based on demographics (for instance, people at both low and high incomes are less likely to answer than those in the middle), but there's really no way to know if that is the full explanation.

$p(M|Y)$ is given as follows:

$$p(M|Y) = p(M|Y_{obs}) \, for \, all \, Y_{mis}. \tag{2}$$

Data are MNAR if the distribution of M depends also on Y_{mis}.

- Simple imputation methods: we distinguish several simple imputation scenarios for the missing data such as:
  - Delete observations containing non-response to one of its characteristics: this method is the most radical one regularly valued. It obliges to apply a filter at first, consisting in setting a limit threshold of non-response per variable and deletes those that exceed it. Then, it deletes all the observations containing a non-response to a single characteristic which involves a loss of information sometimes valuable depending on the size of the data sample. This type of approach raises another point; is there a link between non-response and measured variables?. If such a link is proven, then the approach brings a considerable bias in the analysis of the results.
  - Adjust the analysis according to the different subsets of variables: this method is a variant of the previous one, since the idea is to consider only the variables we need, suppress the observations as described in the previous method and launch the analysis on this subpopulation. For example, if we apply a differential filter, each variable will be considered with its maximum number of observations. If the approach makes it possible to consider the information to the maximum of its availability, it can raise a problem of considerable comparability.
  - Transform the variable: it is a question of reconsidering the analysis from a different admitting angle. For a qualitative variable, a new modality will constitute it to represent the non-response. For a quantitative variable, we code the variable into several modalities, one of which represents the non-response.
    However, this approach requires a relatively high non-response rate to statistically represent an interpretable information.
  - Imputation by the mean, median and the most frequent value: this method allows the insertion in the blank field of an average value derived from the reporting units displaying the same set of predetermined characteristics. For example, if an income is missing from a registration, we could impute the average income of the same province for the same occupation and experience. This method belongs to univariate approaches, based only on the variable under consideration and interaction with the other variables in the dataset.

For quantitative variables, with the classical approach and when we want to impute non-response, we can replace it by the mean of the distribution if the sample size is particular or by the median if not. Qualitative variables are imputed by the most popular method.

- Imputation by the conditional mean: it is more complex than imputation by the mean. It takes into account the interactions of the variable to be imputed with the others in the dataset. Two approaches may be preferred:
  - Constructing a multivariate model which will consider, as the response variable, the variable to be imputed and, as explanatory variables, the others. The value to be imputed is then the mean of the prediction.
  - If we have qualitative variables, then it is a matter of calculating the modal group average of the qualitative variables and averaging the values of the variable to be imputed in order to obtain the imputation value.

- Deterministic imputation methods: deterministic imputation is based on a deterministic model of the type shown by Eq. (3).

$$Y_i = a + b \times X_i + U_i \tag{3}$$

where:

- Y_i is the variable trying to predict.
- X_i is the variable using to predict Y_i.
- a is the intercept.
- b is the slope.
- U_i is the regression residual.

The main defect we can find in this type of approach is that two individuals with the same characteristics will have the same imputation, which is really difficult to justify.

Among the most common deterministic imputation methods we can cite :

- *By decision* : in a first step, we must define a subset of variables for which we have no non-response. The method consists in applying one of the following three algorithms: *k-Nearest Neighbors* [28], *Decision Tree* [22] and *Random Forest* [34].
- *Local regression*: the *LOcal regrESSion* (LOESS) allows missing data to be imputed. For this, a weak polynomial degree is adjusted around the missing data by weighted least squares, giving more weight to the values close to the missing data [7].
- *Nonlinear Iterative Partial Least Squares (NIPALS)*: is an iterative method close to the Partial Least Squares (PLS) regression, used to estimate the elements of a PCA of a random vector of finite dimension. This algorithm can be adapted for imputation of missing data [32].
- *By decomposition into singular values (SVD)*: if there are more observed data than missing data, the dataset Y is separated into two groups: Y^c for the complete observations and Y^m including individuals for whom some data are missing. We consider after that the decomposition into singular values (SVD) truncated from the full dataset [16].

IV. DISTRIBUTED APPROACH FOR HIGH-DIMENSIONALITY HETEROGENOUS DATA REDUCTION

In this section, we introduce a new distributed approach to reduce high-dimensional heterogenous data.

A. MODEL OVERVIEW

Our proposed approach is based essentially on 3 primordial steps as following:

- Data preparation and cleaning;
- Data splitting;
- Data Reduction.

The third step is entirely parallelized according to MapReduce paradigm. Hadoop is the software infrastructure that allow to run our MapReduce schema in a massively distributed way on a cluster of machines, while taking in consideration the issues of distributed computing.

Figure 2 shows the general overview of the proposed approach. As we see Big Data is the feed source of our approach. Because of their heterogeneous nature, these later need to be prepared and cleaned. After that, they will be splitted in order to be processed by the different maps according to their nature to know quantitative or qualitative. Thereafter, the resulting PCA matrix will be the input of the Reduce task to generate as a final output the cleaned and reduced data.

B. PROPOSED ALGORITHM

An algorithmic description detailing the Main procedure is given in algorithm 1. We note here that the execution system is composed of one master, a mapper and a reducer. The Mapper is responsible for the execution of the map function, the reducer executes the reduce function and the master schedules the execution of parallel tasks.

Algorithm 1 Distributed High-Dimensionality Reduction of Heterogeneous Data Algorithm

1: Data cleaning and preparation.
2: Split the quantitative and the qualitative variables.
3: Data Reduction (See Algorithm2)

The main procedure makes it possible to create a Hadoop configuration object required to:

- Allow Hadoop to obtain the general configuration of the cluster. The object in question could also allow the user to obtain the configuration options.
- Allow Hadoop to retrieve possible generic arguments available on the command line (for example, the package name of the task to be executed if the .jar contains more than one). the user will also retrieve the additional arguments to use it.

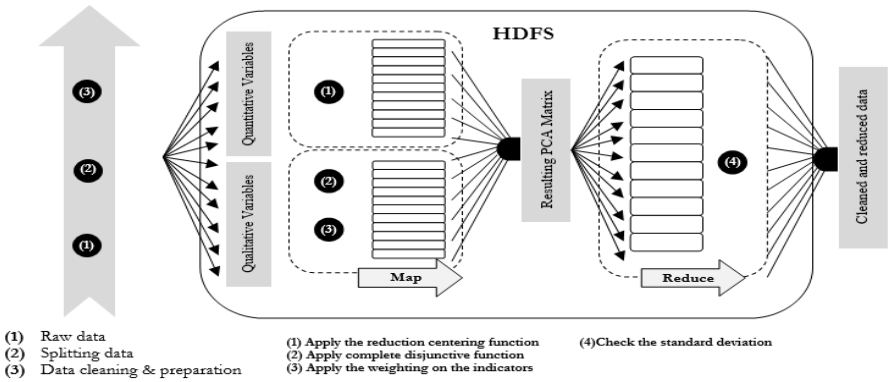

FIGURE 2. General overview of the proposed approach.

The user can specify the name of the input file and the name of the HDFS output directory for the Hadoop tasks using the command line. Thus, a new Hadoop Job object was created (Hadoop task) which in turn informs Hadoop of the names of Main, Map and Reduce tasks. We use the same object to inform Hadoop of the data types used in the program for the $\langle key, value \rangle$ couples. The previously Job Object created is used to trigger the task launch via the Hadoop cluster.

Algorithm 2 The Data Reduction MapReduce Algorithm

Input: X ($n \times p$ matrix) : Data cleaned, prepared and splitted, specifying mappers and reducers numbers.

Output: X' ($n \times p'$ matrix) : Reduced data

//Map Task

1: Apply the reduction centering function on the quantitative variables.

2: Apply the complete disjunctive function on the qualitative variables.

3: Apply the weighting on the indicators.

4: Combine the two matrix resulting from steps 2 and 4 (combination by columns) and rounding the values of the new matrix to 3 decimal digits.

5: Perform a PCA on the given data matrix and returning the results as an object of class, then checking the rotation.

//Reduce Task

6: Check the standard deviation of principal components and compute the proportion variance of each component.

7: Check the proportion variance of components that contain more than 90% of the information.

The input is the Big Data which is of messy heterogenous nature and with missing values. After going through the data preparation and cleaning step and the data splitting ones, the main algorithm calls the Data Reduction step (see algorithm 2). This latter gives us the hand to indicate the number of mappers as well as reducers when entering to the mapreduce programmation paradigm. The quantitative data and the qualitative ones are each processed separately in a Map task. After that, they will be combined to give birth to the resulting PCA matrix, which, in turn, will be the feed source of the Reduce task. Finally, after checking the standard deviation of principal components and compute the proportion variance of each component and checking the proportion variance of components that contain more than 90% of the information, we obtain the cleaned and reduced expected data.

C. APPROACH ILLUSTRATION EXAMPLE

To highlight this approach via an illustrative example, we have used the dataset which describes a sample of 27 small cars from the Belgian market. In this example, we have 9 mixed variables formed by 6 continuous quantitative variables: the displacement, the urban consumption, the maximum speed, the volume of the boot, the weight power ratio and the length. In addition, there are 3 nominal qualitative characteristics : the tax power (4CV, 5CV, 6CV), the manufacturer's mark (French, Foreign) and four price classes (CP1, CP2, CP3, CP4). We have a totaling of 9 modalities as shown in the Table 5.

TABLE 5. Raw data relating to the characteristics of cars.

Name	*Cons*	*Cycli*	*Spreed*	*Volu*	*RP/P*	*Long*	*Tax*	*Marq*	*Price*
AS2	6.20	998	140	955	23.20	3.40	4CV	ETRA	CP1
CI4	560	954	145	1170	1940	350	4CV	FRAN	CP1
PE6	6.70	993	145	1151	20.80	3.61	4CV	FRAN	CP2
FI3	6.30	999	140	1088	21.80	3.64	4CV	ETRA	CP1
FI5	6.20	999	145	968	21.50	3.64	4CV	ETRA	CP2
FI8	8.90	1301	200	968	11.00	3.64	6CV	ETRA	CP4
FID	7.70	1302	165	968	16.00	3.64	6CV	ETRA	CP3
FO1	7.00	1117	137	900	22.70	3.64	4CV	ETRA	CP1
RE7	9.30	1597	180	973	12.00	3.64	6CV	FRAN	CP4
NI1	6.40	988	140	375	17.00	3.64	4CV	ETRA	CP1
OP1	7.20	993	143	845	22.40	3.62	4CV	ETRA	CP1
PE1	6.80	954	134	1200	23.80	3.70	4CV	FRAN	CP1
PE3	5.80	1124	142	1200	21.40	3.70	5CV	FRAN	CP3
DA2	9.20	1360	170	1200	13.90	3.70	6CV	ETRA	CP3
PE9	8.70	1580	190	1200	11.20	3.70	6CV	FRAN	CP4
RE1	6.30	956	115	950	33.10	3.67	4CV	FRAN	CP1
RE3	6.30	1108	120	950	28.40	3.67	5CV	FRAN	CP2
RE4	5.80	1108	143	915	20.60	3.59	5CV	FRAN	CP2
FO9	7.90	1397	167	915	13.80	3.59	6CV	ETRA	CP3
RE8	8.70	1397	200	915	10.20	3.59	6CV	FRAN	CP4
SE4	8.80	1461	175	1200	14.70	3.63	6CV	ETRA	CP3
SE9	7.30	903	131	1088	23.40	3.46	4CV	ETRA	CP1
SZ2	6.40	993	145	400	18.40	3.58	4CV	ETRA	CP1
SZ3	6.50	1324	163	400	14.00	3.58	5CV	ETRA	CP3
TO1	6.10	999	150	202	19.50	3.70	4CV	ETRA	CP2
TO3	6.80	1295	170	202	15.00	3.70	5CV	ETRA	CP3
VW3	7.80	1272	170	1040	14.30	3.65	6CV	ETRA	CP3

1) DATA CLEANING AND PREPARATION

The data cleaning and preparation step is shown in procedure 1.

Procedure 1 *CleanAndPrepare ()*
if *Data contain missing values* **then**
1 Exploring raw data
2 Tidying data
3 Imputing missing values with RandomForest algorithm [34]
end
End

We note that our dataset is tidy because it is clear that 1) each variable forms a column, 2) each observation forms a row and 3) each type of observational unit forms a table (see Table 5).

We draw attention to the fact that despite the increasing of the data amount, missing data problems remain widespread in statistical problems and require a particular approach. Since our approach aims to reduce this deluge of data, we propose to apply a missing data imputation algorithm to our mixed dataset after seeding 10% of missing values on the mixed dataset as shown in Table 6.

TABLE 6. Raw data relating to the characteristics of cars with 10% of missing values.

Name	*Cons*	*Cycli*	*Spreed*	*Volu*	*RP/P*	*Long*	*Tax*	*Marq*	*Price*
1	6.2	998	140	955	23.2	3.40	4CV	ETRA	CP1
2	5.6	954	145	1170	NA	NA	4CV	FRAN	CP1
3	6.7	993	145	1151	20.8	3.61	4CV	FRAN	CP2
4	6.3	999	140	1088	21.8	3.64	4CV	ETRA	CP1
5	6.2	999	145	968	21.5	3.64	4CV	ETRA	CP2
6	8.9	1301	200	968	11.0	3.64	NA	ETRA	CP4
7	7.7	1302	165	NA	16.0	3.64	6CV	ETRA	CP3
8	7.0	1117	137	900	NA	3.64	4CV	ETRA	CP1
9	9.3	1597	180	973	12.0	3.64	6CV	NA	CP4
10	6.4	988	140	375	17.0	3.64	4CV	ETRA	CP1
11	7.2	993	NA	845	22.4	3.62	NA	ETRA	CP1
12	6.8	NA	134	NA	NA	3.70	4CV	FRAN	CP1
13	5.8	1124	142	NA	21.4	NA	5CV	FRAN	CP3
14	9.2	1360	NA	1200	13.9	3.70	6CV	ETRA	NA
15	8.7	NA	190	1200	11.2	3.70	6CV	FRAN	CP4
16	6.3	956	115	950	33.1	3.67	4CV	FRAN	NA
17	6.3	1108	120	950	28.4	3.67	5CV	FRAN	CP2
18	5.8	1108	143	915	20.6	3.59	5CV	FRAN	CP2
19	7.9	1397	167	915	13.8	3.59	6CV	ETRA	CP3
20	8.7	1397	200	915	10.2	3.59	6CV	FRAN	CP4
21	8.8	1461	175	1200	14.7	3.63	NA	ETRA	CP3
22	7.3	903	131	NA	23.4	3.46	4CV	ETRA	NA
23	6.4	993	145	400	18.4	NA	4CV	ETRA	CP1
24	6.5	1324	163	400	14.0	3.58	5CV	ETRA	CP3
25	6.1	999	150	202	19.5	NA	4CV	NA	CP2
26	6.8	1295	NA	202	15.0	3.70	5CV	ETRA	CP3
27	7.8	1272	170	1040	14.3	3.65	6CV	ETRA	CP3

The 10% of missing values added to our initial dataset are broken down as shown in Figure 3.

To impute missing values, we propose to apply the RandomForest algorithm [34]. This algorithm aims to predict individual missing values accurately rather than take random draws from a distribution, so the imputed values may lead to biased parameter estimated in statistical models. We assume $X = (X_1, X_2, \ldots, X_P)$ to be a $n \times p$ dimensional data matrix. The dataset can be separated into four parts:

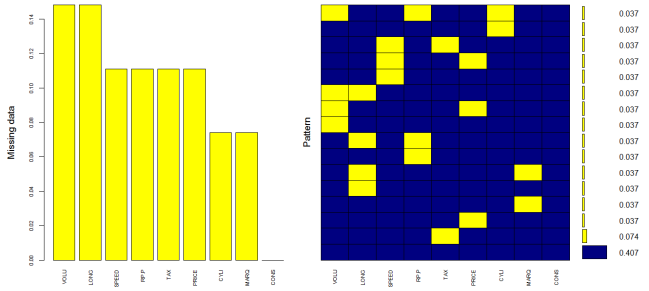

FIGURE 3. Repartition of missing values.

1) The observed values of variable X_s, denoted by $y_{obs}^{(s)}$;
2) the missing values of variable X_s, denoted by $y_{mis}^{(s)}$;
3) the variables other than X_s with observations $i_{obs}^{(s)} = \{1 \ldots n\} \backslash i_{mis}^{(s)}$ denoted by $x_{obs}^{(s)}$; and
4) the variables other than X_s with observations $i_{mis}^{(s)}$ denoted by $x_{mis}^{(s)}$

The imputation procedure is repeated until a stopping criterion is met. The algorithm 3 gives a representation of the missForest method.

Algorithm 3 RandomForest Algorithm
Input : $\mathbf{X}$: an $n \times p$ matrix, stopping criterion γ
Output: The imputed matrix X^{imp}
Make initial guess for missing values
$k \leftarrow$ vector of sorted indices of columns in $\mathbf{X}$
while *not* γ **do**
  $X_{old}^{imp} \leftarrow$ store previously imputed matrix;
  for *s in k* **do**
    Fit a random forest: $y_{obs}^{(s)} \sim x_{obs}^{(s)}$;
    Predict $y_{mis}^{(s)}$ using $x_{mis}^{(s)}$;
    $X_{new}^{imp} \leftarrow$ update imputed matrix using predicted $y_{mis}^{(s)}$;
  end
  Update γ;
end
End

The stopping criterion γ is reached as soon as the difference between the matrix of newly imputed data and the previous one increases for the first time.

The difference of the set of continuous variables N is defined as shown in the Eq. (4).

$$\Delta_N = \frac{\sum_{j \in N} (X_{new}^{imp} - X_{old}^{imp})^2}{\sum_{j \in N} (X_{new}^{imp})^2} \tag{4}$$

In the case of qualitative variables F, the difference will be as described in the Eq. (5).

$$\Delta_F = \frac{\sum_{j \in F} \sum_{i=1}^{n} \mathbb{1}_{X_{new}^{imp} \neq X_{old}^{imp}}}{\#NA} \tag{5}$$

where #NA is the number of missing values in the categorical variables.

After applying the RandomForest algorithm, the imputation of missing values produces the result shown in the Table 7.

TABLE 7. Overview of the cars dataset after imputation of missing values.

Obs	*Cons*	*Cycli*	*Speed*	*Volu*	*RP/P*	*Long*	*Tax*	*Marq*	*Price*
1	6.2	998.00	140.0000	955.000	23.20000	3.4000	4CV	ETRA	CP1
2	5.6	954.00	145.0000	1170.000	21.68600	3.5807	4CV	FRAN	CP1
3	6.7	993.00	145.0000	1151.000	20.80000	3.6100	4CV	FRAN	CP2
4	6.3	999.00	140.0000	1088.000	21.80000	3.6400	4CV	ETRA	CP1
5	6.2	999.00	145.0000	968.000	21.50000	3.6400	4CV	ETRA	CP2
6	8.9	1301.00	200.0000	968.000	11.00000	3.6400	6CV	ETRA	CP4
7	7.7	1302.00	165.0000	848.980	16.00000	3.6400	6CV	ETRA	CP3
8	7.0	1117.00	137.0000	900.000	21.65800	3.6400	4CV	ETRA	CP1
9	9.3	1597.00	180.0000	973.000	12.00000	3.6400	6CV	ETRA	CP4
10	6.4	988.00	140.0000	375.000	17.00000	3.6400	4CV	ETRA	CP1
11	7.2	993.00	138.7217	845.000	22.40000	3.6200	4CV	ETRA	CP1
12	6.8	1007.47	134.0000	933.985	24.63692	3.7000	4CV	FRAN	CP1
13	5.8	1124.00	142.0000	875.095	21.40000	3.5960	5CV	FRAN	CP3
14	9.2	1360.00	174.9700	1200.000	13.90000	3.7000	6CV	ETRA	CP3
15	8.7	1443.33	190.0000	1200.000	11.20000	3.7000	6CV	FRAN	CP4
16	6.3	956.00	115.0000	950.000	33.10000	3.6700	4CV	FRAN	CP1
17	6.3	1108.00	120.0000	950.000	28.40000	3.6700	5CV	FRAN	CP2
18	5.8	1108.00	143.0000	915.000	20.60000	3.5900	5CV	FRAN	CP2
19	7.9	1397.00	167.0000	915.000	13.80000	3.5900	6CV	ETRA	CP3
20	8.7	1397.00	200.0000	915.000	10.20000	3.5900	6CV	FRAN	CP4
21	8.8	1461.00	175.0000	1200.000	14.70000	3.6300	6CV	ETRA	CP3
22	7.3	903.00	131.0000	845.695	23.40000	3.4600	4CV	ETRA	CP1
23	6.4	993.00	145.0000	400.000	18.40000	3.6261	4CV	ETRA	CP1
24	6.5	1324.00	163.0000	400.000	14.00000	3.5800	5CV	ETRA	CP3
25	6.1	999.00	150.0000	202.000	19.50000	3.6137	4CV	ETRA	CP2
26	6.8	1295.00	161.1600	202.000	15.00000	3.7000	5CV	ETRA	CP3
27	7.8	1272.00	170.0000	1040.000	14.30000	3.6500	6CV	ETRA	CP3

To check the imputation error, we have used two concepts: NRMSE (NoRmalized Mean Squared Error) [31] and PFC (Proportion of Falsely Classified) [23]. The NRMSE is used to represent error derived from imputing continuous values whereas PFC is used to represent error derived from imputing categorical values. Table 8 presents the imputation error.

TABLE 8. Imputation error.

	NRMSE	*PFC*
Imputation error	0.2271597	0.1250000

The obtained result suggests that categorical and continuous variables are imputed respectively with 12% and 22% errors. This can be improved by tuning the values of *mtry* which refers to the number of variables being randomly sampled at each split and *ntree* parameter that refers to number of trees to grow in the forest.

2) SPLITTING QUANTITATIVE AND QUALITATIVE VARIABLES

Mixed variables may have different distributions in a dataset:

- All quantitative variables precede all qualitative variables.
- All qualitative variables precede all quantitative variables.
- Random distribution of different types of variables.

The panorama of the quantitative and the qualitative variables is illustrated respectively in tables 9 and 10.

3) DATA CENTERING REDUCTION FOR QUANTITATIVE VARIABLES

In this step, we apply a centering reduction function on the quantitative variables. The reduced center variable or z-score allows to indicate how many standard deviations, above or below the mean, is a sample of data series. To find the reduced centered variable of an item in the sample, we will need to find the mean, variance and standard deviation of the sample. Then we will have to differentiate between the value of this element and the sample mean and finally divide that result by the standard deviation of the same sample.

TABLE 9. Quantitative variables..

Quantitative Variable	*Description*
Cons	The urban consumption
Cycli	The car engine size
Speed	The maximum speed
Volu	The volume of the boot
RP/P	The weightpower ratio
Long	The car length

TABLE 10. Quantitative variables.

Qualitative Variable	*Description*
Tax	The tax power
Marq	The manufacturer's brand
Price	The price

Function 1 describes the main steps of data centering reduction.

Function 1 `DataCenteringReduction ()`

```
if
Quantitative variables x neither centered nor reduced
then
1   n ← length(x)
2   m ← mean(x)
3   v ← (n − 1)/n ∗ var(x)
4   x′ ← (x − m)/sqrt(v)
end
5   return x′
End
```

where :

- $length(x)$ is the number of the quantitative variables which is equal to 6 in our cars case study.
- $mean(x)$ is the arithmetic mean of each quantitative column. It is presented by Eq. (6).

$$\overline{x} = \frac{1}{length(x)} \sum_{i=0}^{length(x)} x_i. \tag{6}$$

- $var(x)$ is a measure of how the quantitative values is dispersed around the mean. It is presented by equation (7).

$$\sigma^2 = \frac{1}{length(x)} \sum_{i=0}^{length(x)} (x_i - \overline{x})^2. \tag{7}$$

After applying the data centering and reduction (see function 1), we have the result shown in the Table 15.

4) COMPLETE DISJUNCTIVE CODING FOR QUALITATIVE VARIABLES

This step deals with the application of the complete disjunctive table on the qualitative variables. This table consists of the bursting of a table defined by **n** observations and **q** qualitative variables in a table defined by **n** observations and **p** indicators where **p** is the sum of the number of modalities of **q** variables: each variable **Q (j)** is decomposed into a

sub-table with **q (j)** columns where column **k** contains **1** for the observations corresponding to the **k-th** modality and **0** for the other observations. In our example, *Tax*, *Marq* and *Price* are the qualitatives variables. The result is presented in the table 16.

5) DATA INDICATOR WEIGHTING

The weighting of the data consists in assigning a weight to each individual in the initial dataset. The primary objective is to correct the representativeness of the initial dataset according to certain key variables in order to facilitate the extrapolation of the result [33]. Function 2 describes the course of the indicator weighting.

Function 2 `DataIndicatorWeighting()`
if *x resulted from the complete disjunctive table* **then**
1 $m \leftarrow mean(x)$
end
2 **return** $x/sqrt(m)$
End

where $mean(x)$ is the arithmetic mean of each qualitative column in the complete disjunctive table. By applying this function, we obtain the result presented in Table 17.

6) TRANSFORMING DATA TO PCA

In this step, we combine the resulted table of the centering and reduction function and the complete disjunctive table. This combination is called *combination by columns*. After that, we round the values of this combination to 3 decimal digits and we obtain the result showed in Table 18.

7) PERFORMANCE OF THE PCA AND CHECK OF THE ROTATION

Here, we apply the PCA method and we select by way of example the first 10 principal components as shown in Table 19.

The rotation measure provides the principal component loading. Each column of rotation matrix contains the principal components loading vector. This is the most important measure we should be interested in.

The rotation returns 15 principal components. Absolutely, in a dataset, the maximum number of principal component loadings is a minimum of $(n-1, p)$. But let firstly look at 6 principal components and first 5 rows (Table 11).

TABLE 11. A view on rotation.

Char/Mod	*PC1*	*SPC2*	*SPC3*	*SPC4*	*SPC5*	*SPC6*
Cons	0.38798548	-0.21948347	-0.02409780	-0.08507856	0.069151165	-0.087680837
Cycli	0.40694841	0.13917699	-0.01334523	-0.07046903	-0.044250745	0.067063289
Speed	0.40984443	-0.03309400	0.01588184	0.21853488	-0.039275713	0.008930522
Volu	0.09883609	-0.31870977	-0.48521501	-0.45246704	-0.331941257	-0.224568219
RPP	-0.38446507	-0.04061497	-0.18502771	-0.24571377	0.005618747	-0.005582461

8) STANDARD DEVIATION AND VARIANCE PROPORTION OF EACH COMPONENT

The standard deviations of the principal components are presented in Table 12.

TABLE 12. The standard deviation of the principal components.

Principal Component	*Standard Deviation*
PC1	2.313565e+00
PC2	1.384514e+00
PC3	1.326332e+00
PC4	1.013944e+00
PC5	9.572112e-01
PC6	8.913288e-01
PC7	4.922679e-01
PC8	4.527177e-01
PC9	3.128842e-01
PC10	2.849443e-01
PC11	2.225722e-01
PC12	1.349255e-01
PC13	2.284893e-16
PC14	1.447886e-16
PC15	1.264521e-16

We aim to find the components which explain the maximum variance. This is because we want to retain as much information as possible using these components. So, higher is the explained variance, higher will be the information contained in those components.

Table 13 explains more the variance proportion of each component.

TABLE 13. Variance proportion of each component.

Principal Component	*Variance proportion*
PC1	4.295284e+01
PC2	1.538237e+01
PC3	1.411669e+01
PC4	8.250046e+00
PC5	7.352649e+00
PC6	6.375353e+00
PC7	1.944605e+00
PC8	1.644687e+00
PC9	7.855894e-01
PC10	6.515507e-01
PC11	3.975304e-01
PC12	1.460886e-01
PC13	4.189480e-31
PC14	1.682275e-31
PC15	1.283159e-31

As shown in Table 14, the first six components alone contain 94.39% of the information, so it is reasonable to keep only six variables by losing only 5.61% of information. We can also account for these proportions with a bar plot of variances. Consequently, our initial dataset may be reduced, instead of 10 variables we can use only 6.

V. EXPERIMENTAL STUDY

A. EXPERIMENTAL PROTOCOL

The master node works as the user interface and hosts both RHadoop master processes: the NameNode and the JobTracker. The NameNode handles the HDFS, coordinating the slave machines by the means of their respective DataNode processes, keeping track of the files and the replications of each HDFS block. The JobTracker is the MapReduce framework master process that manages the TaskTrackers of each compute node. Its responsibilities are maintaining the load-balance and the fault-tolerance in the system, ensuring that all nodes get their part of the input data chunk and reassigning the parts that could not be executed.

TABLE 14. Final result.

Observation	*PC1*	*PC2*	*PC3*	*PC4*	*PC5*	*PC6*
1	-1.8634984	0.62863573	-1.0418581	-0.5077903	0.268017009	0.47682480
2	-1.7943388	1.52930915	-0.9500959	0.8343976	-1.820905314	-0.14233152
3	-1.5573920	1.06191595	1.4059460	0.2872536	0.321695774	-1.40447950
4	-1.8325271	0.08612486	0.4524731	-0.1496600	1.334110458	-1.54084496
5	-1.7789906	-0.14197082	0.3163260	0.1590129	1.389242046	-1.42018380
6	3.7541412	0.93546808	-0.6375156	0.9623834	0.735230911	-0.13684546
7	1.7557867	-0.75497470	-0.5660442	-1.3319598	-0.007295095	-0.46128836
8	-1.5487112	0.53291931	-0.9808807	-0.5496643	0.384421001	0.66707011
9	3.7290534	1.81215807	0.8722000	0.8073841	-0.082709682	0.62377449
10	-1.1794821	-0.61572806	-1.7249990	0.9308946	0.601705259	1.09068719
11	-1.7926302	0.49469580	-1.2104776	-0.2458851	0.223820606	0.48164071
12	-1.7964143	1.53089556	0.8603056	-1.0110280	0.392036423	1.33136343
13	-0.7310097	-1.27210794	2.5450102	-1.3102045	-1.154315654	0.85623836
14	2.8511452	-0.29874017	-0.1680174	-2.0505323	0.625939879	-0.24759491
15	4.1617659	1.62410095	1.2926957	0.4784624	0.584389278	0.81682170
16	-3.2997173	1.40213546	0.8873163	-0.9881229	0.033352917	1.58555046
17	-1.7966755	-0.63350250	3.1159326	0.3669293	-0.253496172	-0.19009400
18	-1.1992244	-0.53618503	2.2390712	1.0948339	-1.180878220	-1.05616749
19	2.1409717	-0.71528156	-1.0083303	-0.9164045	-0.596040123	-0.85452330
20	3.8320507	1.60886982	0.1852774	1.6970697	-0.523829437	0.02561547
21	2.3382666	-0.88381692	-0.6572083	-1.0999923	-0.127702057	-0.47952792
22	-2.2504669	1.65899310	-1.9804787	-0.1975959	-1.734773658	-1.21670943
23	-1.7783053	-0.26247069	-2.0413212	1.0111894	-0.105519346	0.51231640
24	0.5426672	-2.92854339	-0.5208160	0.8801393	-1.629431511	0.03416729
25	-1.6849693	-1.62555913	-0.1689896	1.5191202	2.341765561	-0.13421185
26	0.7894750	-3.54528448	0.0130062	0.7335525	-0.130682800	1.29378535
27	1.9890296	-0.69205643	-0.5285279	-1.4037830	0.111851948	-0.51105327

RHadoop is a complete set where we can process our data efficiently, perform some meaningful analysis [15]. It allows data scientists familiar with R to quickly utilize the enterprise-grade capabilities of the MapR Hadoop distribution directly with the analytic capabilities of R. In fact, it is considered as a collection of three R packages: ***rmr***, ***rhdfs*** and ***rhbase***. The ***rmr*** package requires Rcpp, RJSONIO, bitops, digest, functional, stringr, plyr, reshape2. ***Rhdfs*** require rJava package. We need to put in these packages before install rmr and rhdfs severally. Rmr package provides Hadoop MapReduce practicality R, rhdfs HDFS provides file management and rhbase provides HBase management from among R. Below mentioned packages provides the practicality of Hadoop among R. The figure 4 illustrates the RHaddop architecture.

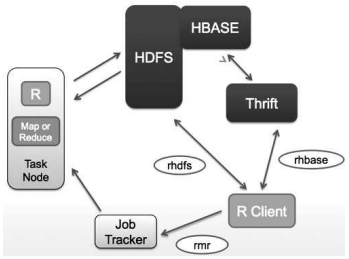

FIGURE 4. Overview of the RHadoop architecture.

- ***Rmr2*** : functions providing Hadoop MapReduce functionality in R.
- ***Rhdfs*** : functions providing file management of the HDFS from within R.
- ***Rhbase*** : functions providing database management for the HBase distributed database from within R.

The specific details of the software used are the following:

- MapReduce implementation: Hadoop 2.6.0-cdh5.5.0. MapReduce 1 runtime(Classic). Cloudera's open-source Apache Hadoop distribution [3].
- Maximum maps tasks: 32, Maximum reducer tasks: 1.
- Operating system: Cent OS 7.0, R x64 3.2.5 version.

TABLE 15. Data centering and reduction of quantitative variables.

Obs	*Cons*	*Cycli*	*Speed*	*Volu*	*RPP*	*Long*
1	-0.3958953	-0.8539321	-0.76880337	0.1728404	0.83109706	-0.07579889
2	-0.3798755	-1.0722543	-0.52618510	-0.1739925	0.11279935	-2.27846313
3	-0.4749927	-0.8787414	-0.52618510	0.9335245	0.37743535	-0.37064371
4	-0.8754862	-0.8489702	-0.76880337	0.7172260	0.56646106	0.14967068
5	-0.9756096	-0.8489702	-0.52618510	0.3052288	0.50975335	0.14967068
6	1.7277216	0.6495141	2.14261581	0.3052288	-1.47501666	0.14967068
7	0.5262411	0.6544760	0.44428796	0.3052288	-0.52988809	0.14967068
8	-0.1746226	-0.2634697	-0.91437433	0.0717638	0.73658421	0.14967068
9	2.1282151	2.1182273	1.17214275	0.3831306	-0.49435125	0.14967068
10	-0.7753629	-0.3048021	-0.08947222	-1.7307237	-0.34086237	0.14967068
11	-0.4099125	-0.8787414	-0.62323241	-0.1170682	0.67987649	-0.19720558
12	-0.3748693	-1.0722543	-0.69128683	1.1017566	0.94451249	1.19029946
13	-1.3761031	-0.2287366	-0.67175606	1.1017566	0.49085078	1.19029946
14	2.0280917	0.9422644	0.68690622	1.1017566	-0.92684209	1.19029946
15	1.5274749	2.0338755	1.65737928	1.1017566	-1.43721152	1.19029946
16	-0.8754862	-1.0623306	-1.98189469	0.2293527	2.70245164	0.66998507
17	-0.8754862	-0.3081265	-0.71118153	0.2434293	1.81403078	0.66998507
18	-0.5510865	-0.3081265	-0.62323241	0.1232634	0.33963020	-0.71751997
19	0.7264878	1.1258535	0.54133526	0.1232634	-0.94574466	-0.71751997
20	1.5274749	1.1258535	2.14261581	0.1232634	-1.62623724	-0.71751997
21	0.4721744	1.4434131	0.92952449	0.2475149	-0.77562152	-0.02376745
22	0.1257475	-1.3253096	-1.20551624	0.7172260	0.86890221	-2.97221565
23	-0.7753629	-0.8787414	-0.52618510	-1.6448909	-0.07622637	-0.89095810
24	-0.6752395	0.7636371	0.34724065	-1.6448909	-0.90793952	-0.89095810
25	-1.0757330	-0.8489702	-0.28356684	-2.3246862	0.13170192	1.19029946
26	-0.3748693	0.6197429	0.68690622	-2.3246862	-0.71891380	1.19029946
27	0.6263644	0.5056199	0.68690622	0.5524271	-0.85123180	0.32310881

B. DATA OF EXPERIMENTATION

The experiments have been yielded on datasets obtained from the University of California at Irvine (UCI) Machine Learning Repository [2]. More details are given as follows:

- Internet Advertisements Data: this dataset was constructed by Nicholas Kushmerick [20] and represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occuring in the URL, the image's URL and alt text, the anchor text, and words occuring near the anchor text. The task is to predict whether an image is an advertisement (''ad'') or not (''nonad''). This dataset consists of 10000 quantitative attributes with 1500 instances.
- Amazon Commerce Reviews data: this dataset is derived from the customers reviews in Amazon Commerce Website for authorship identification. Most previous studies conducted the identification experiments for two to ten authors. Data is in raw form and contains columns of data for real variables. It is created by ZhiLiu et al. [26] and cited as example in [35]. We use a version of this dataset with 3279 instances and 1558 mixed attributes.

The basic information of these benchmark datasets is illustrated in table 20. For each dataset, we show the number of examples (#Instances), number of attributes (#Dimension), the type of attributes (#Type) and the number of classes (#ω).

C. PERFORMANCE MEASURES

To verify the efficiency and performance of the proposed approach, we need several types of measures to characterize its abilities and variants. In the following, we briefly describe the considered measures:

- ***Reduction rate***: It measures the reduction of storage requirements achieved by our parallel algorithm via the equation (8).

$$\textit{Reduction Rate} = \frac{\textit{Number of new Principle Components} * 100\%}{\textit{Number of initial features}} \tag{8}$$

TABLE 16. Complete disjunctive coding.

Tax.4CV	*Tax.5CV*	*Tax.6CV*	*Marq.ETRA*	*Marq.FRAN*	*Price.CP1*	*Price.CP2*	*Price.CP3*	*Price.CP4*
1	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0
1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1
0	0	1	1	0	0	0	1	0
1	0	0	1	0	1	0	0	0
0	0	1	0	1	0	0	0	1
1	0	0	1	0	1	0	0	0
1	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0
0	0	1	1	0	0	0	1	0
0	0	1	0	1	0	0	0	1
1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	1	0	0
0	1	0	0	1	0	1	0	0
0	0	1	1	0	0	0	1	0
0	0	1	0	1	0	0	0	1
0	0	1	1	0	0	0	1	0
1	0	0	1	0	1	0	0	0
1	0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	1	0
1	0	0	1	0	0	1	0	0
0	1	0	1	0	0	0	1	0
0	0	1	1	0	0	0	1	0

TABLE 17. The weighting of indicators.

Obs	*Tax.4CV*	*Tax.5CV*	*Tax.6CV*	*Marq.ETRA*	*Marq.FRAN*	*Price.CP1*	*Price.CP2*	*Price.CP3*	*Price.CP4*
1	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
2	1.441153	0.00000	0.000000	0.000000	1.732051	1.643168	0.00000	0.000000	0.000000
3	1.441153	0.00000	0.000000	0.000000	1.732051	0.000000	2.32379	0.000000	0.000000
4	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
5	1.441153	0.00000	0.000000	1.224745	0.000000	0.000000	2.32379	0.000000	0.000000
6	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	0.000000	2.598076
7	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
8	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
9	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	0.000000	2.598076
10	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
11	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
12	1.441153	0.00000	0.000000	0.000000	1.732051	1.643168	0.00000	0.000000	0.000000
13	0.000000	2.32379	0.000000	0.000000	1.732051	0.000000	0.00000	1.837117	0.000000
14	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
15	0.000000	0.00000	1.732051	0.000000	1.732051	0.000000	0.00000	0.000000	2.598076
16	1.441153	0.00000	0.000000	0.000000	1.732051	1.643168	0.00000	0.000000	0.000000
17	0.000000	2.32379	0.000000	0.000000	1.732051	0.000000	2.32379	0.000000	0.000000
18	0.000000	2.32379	0.000000	0.000000	1.732051	0.000000	2.32379	0.000000	0.000000
19	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
20	0.000000	0.00000	1.732051	0.000000	1.732051	0.000000	0.00000	0.000000	2.598076
21	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
22	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
23	1.441153	0.00000	0.000000	1.224745	0.000000	1.643168	0.00000	0.000000	0.000000
24	0.000000	2.32379	0.000000	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
25	1.441153	0.00000	0.000000	1.224745	0.000000	0.000000	2.32379	0.000000	0.000000
26	0.000000	2.32379	0.000000	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000
27	0.000000	0.00000	1.732051	1.224745	0.000000	0.000000	0.00000	1.837117	0.000000

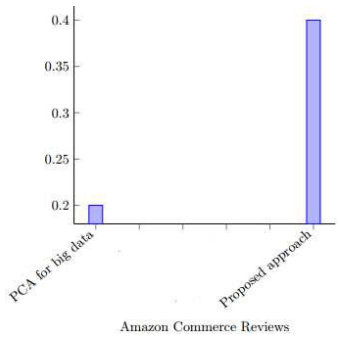

FIGURE 5. Accuracy measures: Spearman coefficient.

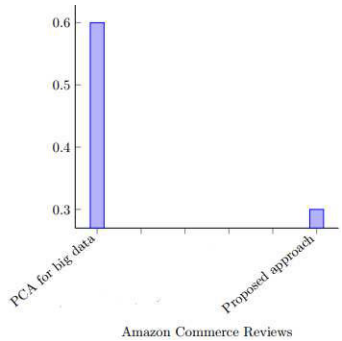

FIGURE 6. Accuracy measures: Shannon Entropy.

- ***Accuracy***: We use two different measures to evaluate the accuracy:
  - *Spearman coefficient* (The Spearman's Rank Correlation Coefficient). It is a statistical measure used to discover the strength of a link between two sets of data [17]. This ratio uses the ranks of variables instead of their values. Possible values range from -1 (strong negative relation) to 1 (strong positive relation). If ratio is equal to zero, this means there is no statistical link between datasets. To calculate this ratio, R function cor() with method "spearman" was used.

TABLE 18. Transformed data input of the PCA.

Cons	*Cycli*	*Speed*	*Volu*	*RP.P*	*Long*	*Tax.4CV*	*Tax.5CV*	*Tax.6CV*	*Marq.ETRA*	*Marq.FRAN*	*Price.CP1*	*Price.CP2*	*Price.CP3*	*Price.CP4*
-0.396	-0.854	-0.769	0.173	0.831	-0.076	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
-0.380	-1.072	-0.526	-0.174	0.113	-2.278	1.441	0.000	0.000	0.00	1.643	1.732	0.000	0.000	0.000
-0.475	-0.879	-0.526	0.934	0.377	-0.371	1.441	0.000	0.000	0.00	1.643	0.000	2.121	0.000	0.000
-0.875	-0.849	-0.769	0.717	0.566	0.150	1.441	0.000	0.000	1.26	0.000	0.000	2.121	0.000	0.000
-0.976	-0.849	-0.526	0.305	0.510	0.150	1.441	0.000	0.000	1.26	0.000	0.000	2.121	0.000	0.000
1.728	0.650	2.143	0.305	-1.475	0.150	0.000	0.000	1.732	1.26	0.000	0.000	0.000	0.000	2.598
0.526	0.654	0.444	0.305	-0.530	0.150	0.000	0.000	1.732	1.26	0.000	0.000	0.000	1.837	0.000
-0.175	-0.263	-0.914	0.072	0.737	0.150	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
2.128	2.118	1.172	0.383	-0.494	0.150	0.000	0.000	1.732	0.00	1.643	0.000	0.000	0.000	2.598
-0.775	-0.305	-0.089	-1.731	-0.341	0.150	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
-0.410	-0.879	-0.623	-0.117	0.680	-0.197	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
-0.375	-1.072	-0.691	1.102	0.945	1.190	1.441	0.000	0.000	0.00	1.643	1.732	0.000	0.000	0.000
-1.376	-0.229	-0.672	1.102	0.491	1.190	0.000	2.324	0.000	0.00	1.643	0.000	0.000	1.837	0.000
2.028	0.942	0.687	1.102	-0.927	1.190	0.000	0.000	1.732	1.26	0.000	0.000	0.000	1.837	0.000
1.527	2.034	1.657	1.102	-1.437	1.190	0.000	0.000	1.732	0.00	1.643	0.000	0.000	0.000	2.598
-0.875	-1.062	-1.982	0.229	2.702	0.670	1.441	0.000	0.000	0.00	1.643	1.732	0.000	0.000	0.000
-0.875	-0.308	-0.711	0.243	1.814	0.670	0.000	2.324	0.000	0.00	1.643	0.000	2.121	0.000	0.000
-0.551	-0.308	-0.623	0.123	0.340	-0.718	0.000	2.324	0.000	0.00	1.643	0.000	2.121	0.000	0.000
0.726	1.126	0.541	0.123	-0.946	-0.718	0.000	0.000	1.732	1.26	0.000	0.000	0.000	1.837	0.000
1.527	1.126	2.143	0.123	-1.626	-0.718	0.000	0.000	1.732	0.00	1.643	0.000	0.000	0.000	2.598
0.472	1.443	0.930	0.248	-0.776	-0.024	0.000	0.000	1.732	1.26	0.000	0.000	0.000	1.837	0.000
0.126	-1.325	-1.206	0.717	0.869	-2.972	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
-0.775	-0.879	-0.526	-1.645	-0.076	-0.891	1.441	0.000	0.000	1.26	0.000	1.732	0.000	0.000	0.000
-0.675	0.764	0.347	-1.645	-0.908	-0.891	0.000	2.324	0.000	1.26	0.000	0.000	0.000	1.837	0.000
-1.076	-0.849	-0.284	-2.325	0.132	1.190	1.441	0.000	0.000	1.26	0.000	0.000	2.121	0.000	0.000
-0.375	0.620	0.687	-2.325	-0.719	1.190	0.000	2.324	0.000	1.26	0.000	0.000	0.000	1.837	0.000
0.626	0.506	0.687	0.552	-0.851	0.323	0.000	0.000	1.732	1.26	0.000	0.000	0.000	1.837	0.000

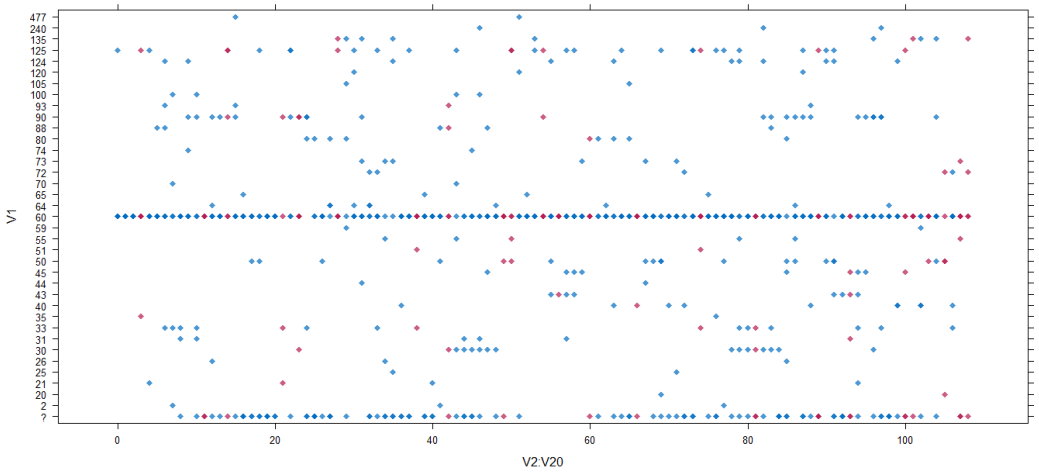

FIGURE 7. Inspecting the distribution of original and imputed data for the *Internet Advertisements dataset*.

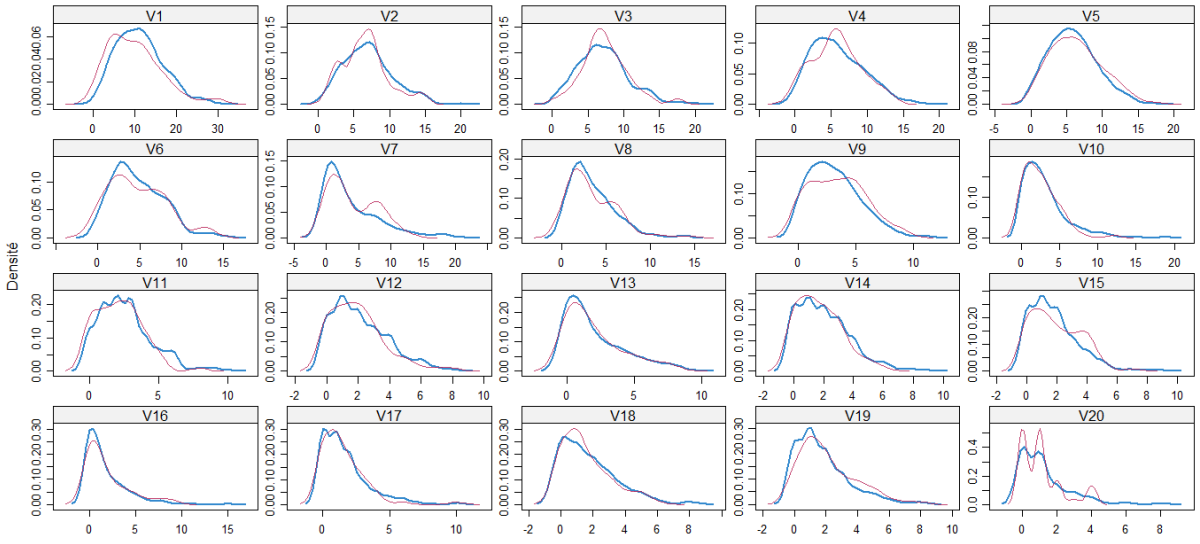

FIGURE 8. Inspecting the distribution of original and imputed data for the *Amazon Commerce Review dataset*.

– *Shannon entropy*: We used R function entropy from package 'entropy' that estimates the Shannon entropy H of the random variable Y from the corresponding observed items [18]. This estimator shows how accurate the projection got by using particular dimensionality reduction method retains

TABLE 19. Performance of the PCA.

Characteristics and modalities	*PC1*	*PC2*	*PC3*	*PC4*	*PC5*	*PC6*	*PC7*	*PC8*	*PC9*	*PC10*
Cons	0.385948204	0.26514478	-0.09391494	-0.08272667	0.003860226	-0.03185104	-0.261505308	0.25321969	0.67508970	-0.01698047
Cycli	0.412719957	-0.09223223	0.07290734	-0.01365197	-0.056100354	0.107335143	-0.076491318	0.42357765	-0.35080524	-0.68381056
Speed	0.418886364	-0.03184501	-0.04755604	0.22587029	0.077649583	-0.006049091	0.087481495	-0.27644127	0.08174822	0.19997007
Volu	0.075041907	0.45070704	0.29673047	-0.56796179	-0.090228414	-0.306357804	-0.161599653	-0.43894309	-0.10893383	-0.18897214
RP.P	-0.378547093	0.14340719	0.20738048	-0.20440230	0.008728414	0.128356159	-0.376424302	0.49539077	-0.17567349	0.32929649
Long	0.081948286	-0.20158813	0.36093142	-0.26324633	0.672744736	0.506502201	0.009403635	-0.13480151	0.13489671	-0.02443909
Tax.4CV	-0.250163711	0.18159104	-0.19462163	0.05888163	0.219535401	0.020002146	0.187721856	-0.11273103	-0.09011255	-0.24716466
Tax.5CV	-0.039987416	-0.41570030	0.37557170	0.15346234	-0.424207048	0.105515882	-0.435998526	-0.21878289	0.20026672	-0.06926543
Tax.6CV	0.330464223	0.09159944	-0.04602853	-0.18515099	0.052336550	-0.102686482	0.099360522	0.29855698	-0.04096820	0.34868235
Marq.ETRA	0.004092276	-0.20551850	-0.34314401	-0.10641009	0.194924299	-0.149250694	-0.353565541	-0.08026986	-0.07796276	-0.01258737
Marq.FRAN	-0.005335675	0.26796334	0.44740505	0.13874179	-0.254150192	0.194599097	0.460993065	0.10465910	0.10165101	0.01641192
Price.CP1	-0.215357170	0.23977438	-0.30986183	-0.04688790	-0.127806329	0.401348338	-0.065992357	-0.04477091	0.24324046	-0.21902062
Price.CP2	-0.150139814	-0.07615374	0.34139008	0.26010351	0.351951330	-0.590094052	0.013790728	0.16929328	0.17498962	-0.14064738
Price.CP3	0.154136846	-0.40881962	-0.03578503	-0.44667779	-0.224230116	-0.032889005	0.277232550	0.01446642	-0.10404774	0.18398528
Price.CP4	0.288936304	0.31176558	0.09728467	0.38346920	0.077768178	0.167204234	-0.309967620	-0.16064332	-0.43203261	0.24059362

TABLE 20. Details of the used datasets.

Datasets	*#Dimension*	*#Type*	*#Instances*	*#ω*
Internet Advertisements	1559	Mixed	3279	2
Amazon Commerce Reviews	10000	Quantitative	1500	50

the initial amount of information. A lesser value of this measure means better accuracy.

- *Test classification time*: It refers to the time needed to classify all the instances in the new dimension space of regarding a given initial dataset.

1) REDUCTION RATE

The first dataset, *Internet Advertisements*, is composed of 1554 quantitative variables and 5 categorical variables. In fact, we sowed 10% of missing values randomly. We imputed these latter by using the missForest algorithm [34] as we have done in out approach. To inspect the distribution of original and imputed data, we used the *stripplot()* function that shows the distribution of the variables as individual points as described in figure 7. What we would like to see is that the shape of the magenta points (imputed) matches the shape of the blue ones (observed). The matching shape tells us that the imputed values are indeed *plausible values*. By applying our algorithm, we arrive to reduce the dimensionality from 1559 mixed variables to the half nearly. Only 778 variables contain about 90.08% of information. We arrived at this important result after a certain time estimated by 9.262416 seconds which is noteworthy.

The second dataset, *Amazon Commerce Reviews*, is a particular case of data type that our algorithm supports because all variables are quantitative. We sowed randomly 10% of missing values in this dataset. In fact, we imputed these latter by using the missForest algorithm [34]. To inspect the distribution of original and imputed data, we used another helpful plot which is the *density plot()*. The density of the imputed data for the 20th imputed variable is showed in magenta while the density of the observed data is showed in blue as shown in figure 8. Again, under our previous assumptions we expect the distributions to be similar.

After applying the new approach of data reducing (which taken 3.638808′), we concluded that only 1300 variables can resume 97.76% of information, which is impressive.

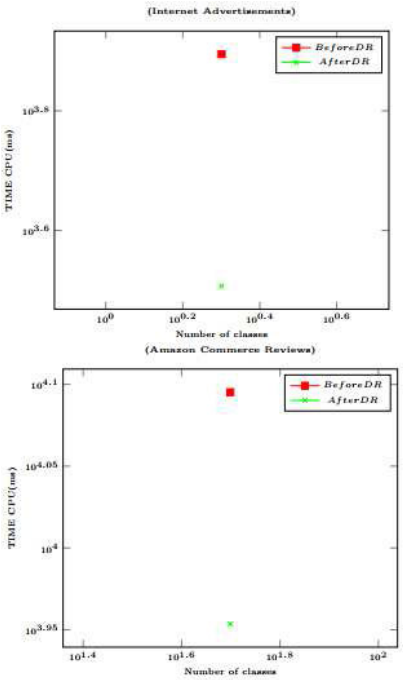

FIGURE 9. The classification time comparaison on ***Internet Advertisements*** and ***Amazon Commerce Reviews***.

2) ACCURACY

In this part, we demonstrate that our DR approach deals with the best accuracy according to the PCA method for big data [42] when processing the *Amazon Commerce Reviews* dataset.

According to the *Spearman coefficient* (Figure 5), the best accuracy is achieved when processing the *Amazon Commerce Reviews* dataset whith our approach.

According to the *Shannon entropy* (Figure 6), there is a significant difference of accuracy between reduced data resulted from the PCA method for big data [42] and those resulted from our method. This estimator shows clearly how accurate the projection got by using particular dimensionality reduction method retains the initial amount of information. A lesser value of this measure when processing our proposed method means better accuracy.

3) ABOUT RUNTIME

In order to enrich the experiment and to more strongly verify the suggested points, a second experiment is conducted

using the same datasets that tests the test classification time. This latter means the time that takes a dataset to reach its conventional classes number. According to figure 9, after applying our algorithm, both datasets reached their optimal classes number in a time less than the one before applying it. That proves that our approach not only reduces the datasets dimension but also keeps practically the same information resides on it.

VI. CONCLUSION

Big Data revolution has led the scientific community to ask questions about infrastructures and architectures capable of handling large volumes of varied data. Thus, many professional and open source solutions appear on the market facilitating the processing and data analysis in large dimensions.

However, when it comes to data analysis, Big Data raises theoretical and statistical problems that must also be addressed. Many classical statistical algorithms are undermined by scaling and problems of robustness and stability arise. Therefore, to limit this curse of dimensionality, we have proposed a distributed high-dimensionality reduction algorithm of heterogeneous data based on the MapReduce paradigm handling by the RHadoop framework.

The suggested algorithm is based on both PCA and MCA which are both descriptive methods. The PCA method enables the processing of quantitative variables while MCA method enables the processing of categorical variables.

Comprehensive experiments on two real datasets have been conducted to study the impact of using our algorithm to limit the curse of dimensionality.

As future work, we propose to i) test our algorithm on our previous contribution in the Big Data Mining context [12]–[14]; ii) use Big data analytics to uncover hidden patterns, unknown correlations and other useful information residing in the huge volume of data and iii) benefit from our distributed approach in favor to enhance the Big Data Mining process.

REFERENCES

[1] M. R. Ahmad, "A significance test of the RV coefficient in high dimensions," *Comput. Statist. Data Anal.*, vol. 131, pp. 116–130, Mar. 2019.

[2] A. Asuncion and D. Newman. (2007). *UCI Machine Learning Repository*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[3] S. A. A. Benaouda, "Implantation du modèle mapreduce dans l'environnement distribué Hadoop: Distribution cloudera," Ph.D. dissertation, 2015.

[4] M. Cannataro, "Big data analysis in bioinformatics," in *Encyclopedia of Big Data Technologies*. Cham, Switzerland: Springer, 2019, pp. 161–180.

[5] M. Cavallo and Ç. Demiralp, "A visual interaction framework for dimensionality reduction based data exploration," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 635:1–635:13.

[6] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: Challenges, issues, and opportunities," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2013, pp. 1–15.

[7] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: An approach to regression analysis by local fitting," *Publications Amer. Statist. Assoc.*, vol. 83, no. 403, pp. 596–610, 1988.

[8] S. Cohen, E. Ruppin, and G. Dror, "Feature selection based on the shapley value," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, 2005, pp. 665–670.

[9] C. Ding, *Dimension Reduction Techniques for Clustering*. Boston, MA, USA: Springer, 2009, p. 846.

[10] I. El Moudden, S. El bernoussi, and B. Benyacoub, "Modeling human activity recognition by dimensionality reduction approach," in *Proc. 27th Int. Bus. Inf. Manage. Assoc. Conf.-Innov. Manage. Educ. Excellence Vis.*, 2016, pp. 1800–1805.

[11] I. El Moudden, M. Ouzir, B. Benyacoub, and S. El Bernoussi, "Mining human activity using dimensionality reduction and pattern recognition," *CES*, vol. 9, no. 21, pp. 1031–1041, 2016.

[12] R. M. Gahar, O. Arfaoui, M. S. Hidri, and N. B. Hadj-Alouane, "Vers une approche heuristique distribuée à base d'ontologie pour la fouille des règles d'association dans les données massives," *Rev. Nouvelles Technol. l'Inf.*, pp. 377–378.

[13] R. M. Gahar, O. Arfaoui, M. S. Hidri, and N. B. Hadj-Alouane, "ParallelCharMax: An effective maximal frequent itemset mining algorithm based on mapreduce framework," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2017, pp. 571–578.

[14] R. M. Gahar, O. Arfaoui, M. S. Hidri, and N. B. Hadj-Alouane, "An ontology-driven mapreduce framework for association rules mining in massive data," in *Proc. 22nd Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst. (KES)*, 2018, pp. 224–233.

[15] D. Harish, M. Anusha, and K. DayaSagar, "Big data analysis usingRHadoop," *Int. J. Innov. Res. Adv. Eng.*, vol. 2, no. 4, pp. 180–185, 2015.

[16] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein, "Imputing missing data for gene expression arrays," Tech. Rep., 1999.

[17] J. Hauke and T. Kossowski, "Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data," *Quaestiones Geograph.*, vol. 30, no. 2, pp. 87–93, 2011.

[18] J. Hausser and K. Strimmer, "Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks," *J. Mach. Learn. Res.*, vol. 10, pp. 1469–1484, Jul. 2009.

[19] S. Jung, "Continuum directions for supervised dimension reduction," *Comput. Statist. Data Anal.*, vol. 125, pp. 27–43, Sep. 2018.

[20] N. Kushmerick, "Learning to remove Internet advertisements," in *Proc. 3rd Annu. Conf. Auto. Agents (AGENTS)*, 1999, pp. 175–181.

[21] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proc. VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, Aug. 2012.

[22] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *Proc. 3rd Annu. Symp. Document Anal. Inf. Retr.*, vol. 33, 1994, pp. 81–93.

[23] S. G. Liao, Y. Lin, D. D. Kang, D. Chandra, J. Bon, N. Kaminski, F. C. Sciurba, and G. C. Tseng, "Missing value imputation in high-dimensional phenomic data: Imputable or not, and how?" *BMC Bioinf.*, vol. 15, no. 1, p. 346, 2014.

[24] R. J. A. Little and D. B. Rubin, "The analysis of social science data with missing values," *Sociol. Methods Res.*, vol. 18, nos. 2–3, pp. 292–326, 1989.

[25] R. J. A. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. Hoboken, NJ, USA: Wiley, 2014.

[26] S. Liu, Z. Liu, J. Sun, and L. Liu, "Application of synergetic neural network in online writeprint identification," *Int. J. Digit. Content Technol. Appl.*, vol. 5, no. 3, pp. 126–135, 2011.

[27] G. Manogaran, D. Lopez, C. Thota, K. M. Abbas, S. Pyne, and R. Sundarasekar, "Big data analytics in healthcare Internet of Things," in *Innovative Healthcare Systems for the 21st Century*. 2017, pp. 263–284.

[28] B. Masand, G. Linoff, and D. Waltz, "Classifying news stories using memory based reasoning," in *Proc. 15th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1992, pp. 59–65.

[29] J. Pagès, "Analyse factorielle de donnees mixtes: Principe et exemple d'application," SupAgro, Montpellier, France, Tech. Rep., 2004.

[30] E. Pierson and C. Yau, "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis," *Genome Biol.*, vol. 16, no. 1, pp. 1–10, 2015.

[31] A. A. Poli and M. C. Cirillo, "On the use of the normalized mean square error in evaluating dispersion model performance," *Atmos. Environ. A, Gen. Topics*, vol. 27, no. 15, pp. 2427–2434, Oct. 1993.

[32] C. Preda, G. Saporta, and M. H. B. H. Mbarek, "The NIPALS algorithm for missing functional data," *Rev. Roumaine Math. Pures Appl.*, vol. 55, no. 4, pp. 315–326, 2010.

[33] G. Saporta, "Pondération optimale de variables qualitatives en analyse des données," *Stat. Anal. Données*, vol. 4, no. 3, pp. 19–31, 1979.

[34] D. J. Stekhoven and P. Bühlmann, "MissForest—Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.

[35] J. Sun, Z. Yang, P. Wang, and S. Liu, "Variable length character n-gram approach for online writeprint identification," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, 2010, pp. 486–490.
[36] A. Taher and A. E. Hassanien, "Dimensionality reduction of medical big data using neural-fuzzy classifier," *Soft Comput.*, vol. 19, no. 4, pp. 1115–1127, 2014.
[37] M. Tenenhaus, "Analyse en composantes principales d'un ensemble de variables nominales ou numériques," *Rev. Stat. Appl.*, vol. 25, no. 2, pp. 39–56, 1977.
[38] M. Tenenhaus and F. W. Young, "An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data," *Psychometrika*, vol. 50, no. 1, pp. 91–119, Mar. 1985.
[39] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[40] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[41] X. Xu, X. Li, and J. Zhang, "Sufficient dimension reduction and prediction through cumulative slicing PFC," *J. Stat. Comput. Simul.*, vol. 88, no. 6, pp. 1172–1190, 2018.
[42] T. Zhang and B. Yang, "Big data dimension reduction using PCA," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, Nov. 2016, pp. 152–157.

RANIA MKHININI GAHAR was born in M'Saken, Sousse, Tunisia. She received the Engineering degree from the Higher Institute of Applied Sciences and Technology of Sousse, Tunisia, in 2013, the master's degree from the National Engineering School of Tunis, Tunisia, in 2015. Since 2016, she has been the Ph.D. student (STIC speciality) and a member of the OASIS Research Lab with the National School of Engineers of Tunis. Her research interests include big data analytics, as well as data science and statistical machine learning.

OLFA ARFAOUI was born in Bou Arada, Seliana, Tunisia. She received the Engineering degree in computer science from the National School of Engineering of Tunis (ENIT), Tunis, EL Manar University, Tunisia, in 2007, and the Ph.D. degree from ENIT, Tunisia, in 2014. She is currently an Assistant Professor with the University of Carthage, Tunisia. Her research activities focus on clustering, XML native databases, flexible querying, big data analytics, as well as data science.

MINYAR SASSI HIDRI was born in Nabeul, Tunisia. She received the Engineering degree in computer science and the Ph.D. degree in computer science from the National Engineering School of Tunis, Tunisia, in 2003 and 2007, respectively, and the Habilitation degree to lead research in computer sciences from Tunis El Manar University, Tunisia, in June 2014. She is currently an Assistant Professor with Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. She has been an Associate Professor with the Computer Sciences Department, National Engineering School of Tunis, Tunis El Manar University, since November 2018. Her experience in teaching, in computer science and information systems is around 16 years. She had worked for five years as a Teaching Assistant at the National Engineering School of Tunis and ten years as an Assistant Professor. Besides her academic responsibilities, she had participated in several administrative tasks such that student's projects and course's coordinator. Her research interests include combinatorial aspects in big data and their applications to different fields, including data mining, machine learning, deep learning, and text mining, with over 65 publications. She is also a member of the steering committee of many international conferences and Reviewer of impacted journals.

NEJIB BEN HADJ-ALOUANE received the B.S. degree in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, and the M.S.E. and Ph.D. degrees in computer information and control engineering from the University of Michigan, Ann Arbor, in 1986 and 1994, respectively. He is currently a Professor with the National School of Engineers of Tunis (ENIT), University of Tunis El Manar, Tunisia. His research interests include discrete-event and hybrid systems, security in computer networks and systems, and issues in real-time and embedded systems, as well as web services and their composition.